

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ АДАПТИВНОГО ФІЛЬТРА-ПРЕДИКТОРА НА АПАРАТНІЙ ОБЧИСЛЮВАЛЬНІЙ ПЛАТФОРМІ ARDUINO

Актуальною задачею сьогодення залишається розробка та впровадження новітніх методів цифрової обробки сигналів (ЦОС). Реалізація таких алгоритмів на основі промислових контролерів і проведення експериментів з їх використання не завжди можливо, так як це вимагає значних економічних ресурсів. Альтернативою для проведення дослідів ефективності роботи методів ЦОС є використання недорогих апаратних обчислювальних платформ. Однією із них є Arduino, основними компонентами якої є плата мікроконтролера і середовище розробки.

Метою даного дослідження була реалізація запропонованого адаптивного фільтра-предиктора[1,2] у середовищі розробки ArduinoIDE, визначення ефективності роботи даного алгоритму в режимі реального часу і особливостей його програмної реалізації. У якості плати керування використано ArduinoUNOз процесором ATmega328P. Програма здійснювала обробку даних, які надходили з ультразвукового далекоміру (УЗД) HC-SR04.

У якості фільтруючої одиниці адаптивного фільтра-предиктора використовується подвійне експоненціальне згладжування. Для реалізації алгоритму у вигляді програми його можна описати за допомогою різницевої рівнянь:

$$\hat{x}(k) = \alpha \cdot x(k) + (1 - \alpha) \cdot \hat{x}(k - 1), \#(1)$$

$$\hat{\hat{x}}(k) = \alpha \cdot \hat{x}(k) + (1 - \alpha) \cdot \hat{\hat{x}}(k - 1), \#(2)$$

$$\hat{\hat{x}}(k + m) = \alpha \cdot (\hat{x}(k) + m \cdot \Delta \hat{x}(k)) + \beta \cdot (\hat{\hat{x}}(k - 1) + m \cdot \Delta \hat{\hat{x}}(k - 1)), \#(3)$$

де $\hat{x}(k)$ – згладжене значення сигналу k -того кроку;

α – коефіцієнт згладжування (ваговий коефіцієнт фільтрації);

$x(k)$ – значення сигналу на вході алгоритму обробки даних (вхідний сигнал) на k -тому кроці;

$(1 - \alpha)$ – другий член ряду нескінченної геометричної прогресії в моделі Брауна, що є ваговим коефіцієнтом другої складової відфільтрованого значення сигналу;

m – кількість кроків, на які необхідно отримати прогноз;

$$\Delta \hat{x}(k) = \hat{x}(k) - \hat{x}(k - 1);$$

$$\Delta \hat{\hat{x}}(k - 1) = \hat{\hat{x}}(k - 1) - \hat{\hat{x}}(k - 2).$$

Рівняння 1, 2 і 3 були реалізовано у вигляді функції, яка дозволила зберігати дані у одновимірні масиви. Поточне значення відфільтрованого сигналу записувалось у перший елемент масиву, а попереднє – у другий. Із кожним наступним викликом даної функції значення першого елемента зміщувалось у другий, після чого перший елемент масиву замінювався оновленим обчисленим значенням. Для функціонування ноніусного

принципу нарощування [1] використано набір одномірних масивів, які зберігали значення відфільтрованого сигналу, відфільтрованих похибок слідування першого і другого порядків.

Відповідно до рівняння 3 визначення прогнозованого сигналу потребує значення $\hat{x}(k-2)$ – двічі згладженого сигналу на кроці $k-2$. Запис даного значення у третій елемент масиву реалізовано наступним чином. У функцію було додано змінну, яка здійснювала роль лічильника c . При виклику функції значення лічильника збільшувалось на 1. Також було додано дві змінні $xkm2_1$ і $xkm2_2$, значення яких на момент початку роботи програми були рівні 0. У функцію було додано перевірку лічильника. Якщо значення $c=1$ то третій елемент масиву приймав значення $xkm2_2$, а змінна $xkm2_1 = \hat{x}(k)$. При наступному виклику функції ($c=2$) третій елемент масиву приймав значення $xkm2_1$, яке в той момент часу відповідало значенню $\hat{x}(k-1)$, а $xkm2_2 = \hat{x}(k)$ і здійснювалось обнуління лічильника. На момент наступного виклику функції при визначенню прогнозованого значення третій елемент масиву відповідав значенню $\hat{x}(k-2)$, після чого даний алгоритм повторювався.

За допомогою функцій *millis()* і *micros()* було визначено час, який необхідний для обчислень, що передбачає алгоритм. Дане значення може бути враховане у процесі вибору частоти отримання даних із датчиків. Так, наприклад, УЗД HC-SR04 для коректного визначення відстані до перешкоди потребує затримку між зчитуваннями даних 33мс. Віднімаючи час роботи алгоритму від даної затримки можна підвищити швидкодію роботи системи.

Висновки: було реалізовано адаптивний фільтр-предиктор на апаратній обчислювальній платформі Arduino; використання функцій і масивів для обробки даних дозволило структурувати і зменшити об'єм програми; визначення часу, який необхідний для обробки даних, дозволило підвищити роботу системи, в яку інтегрується алгоритм ЦОС.

Література

1. Боряк Б.Р. Алгоритм згладжування та прогнозування сигналу на основі експоненціального фільтра моделі Брауна / Б.Р. Боряк, А.М. Сільвестров // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2017. – Т. 4 (44). – С. 150-152.

2. Боряк Б.Р. Дослідження алгоритму згладжування і налаштування адаптивного фільтра Брауна при зміні амплітуди шумів/ Б. Р. Боряк // 69 наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету, квітень-травень, 2017р., - Полтава: ПолтНТУ імені Юрія Кондратюка. Том 1, С – 4-6.